SHORT MESSAGE ENCRYPTION APPLICATION DEVELOPMENT USING VIGENERE ALGORITHM UTILIZING EULER'S NUMBER ON ANDROID SMARTPHONE

Nofiyanto[1], Hamzah[2], Herison Surbakti[3]

Abstract

SMS (Short Message Service) is a facility for sending and receiving a short message of text by using a cell phone. SMS is a medium of communication that is easy to use and relatively cheap for SMS costs. Messages sent via SMS does not guarantee the security and confidential because messages that are sent using the default SMS application for your mobile phone is still an open text that has not been protected. Therefore, it takes a method and applications to secure the confidentiality of the information in an SMS message.

This research aims to develop short message encryption applications on android smartphones using Vigenere algorithm utilizing euler's number as a safety and maintaining the confidentiality of this SMS only to a functioning android smartphone users to send messages and message encryption as well as receive incoming messages and decrypt the message. Starting from the early stages of identifying, needs analysis, design, implementation and testing.

Application of message encryption using vigenere algorithm utilizing euler's number digit can help users of android smartphone to secure the contents of the message are confidential.

Keywords: Short Message Service, the number of euler, vigenere algorithm

[1]Student at Respati University of Yogyakarta
[2]Lecture at Respati University of Yogyakarta
[3]Mentor at Respati University of Yogyakarta

PENDAHULUAN

Perkembangan teknologi di bidang komunikasi semakin tahun semakin maju. Salah satu hasil perkembangan teknologi di bidang komunikasi adalah layanan SMS (*Short Message Service*). SMS merupakan suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks dengan menggunakan telepon seluler. SMS merupakan salah satu media komunikasi yang masih popular di kalangan masyarakat, karena selain mudah digunakan biaya untuk SMS juga tergolong murah.

Melalui layanan SMS, dapat memungkinkan seseorang untuk mengirim dan menerima pesan yang bersifat personal atau rahasia dari orang lain. Namun, pesan yang dikirimkan melalui SMS tidak menjamin keamanan dan kerahasiaannya. Karena pesan yang dikirim menggunakan aplikasi SMS bawaan ponsel masih berupa teks terbuka yang belum terproteksi selain itu pengiriman SMS yang dilakukan tidak sampai ke penerima langsung, akan tetapi pengiriman SMS harus melewati *Short Message Service Center* (SMSC) yang berfungsi mencatat komunikasi yang terjadi antara pengirim dan penerima. Dengan tersimpannya SMS pada SMSC, maka seorang operator dapat memperoleh informasi atau membaca SMS di dalam SMSC tersebut, hal ini dapat dibuktikan dari beberapa kasus yang ditangani pihak kepolisian, kejaksaan atau KPK(Komisi Pemberantasan Korupsi), dimana pihak-pihak tersebut meminta transkip SMS ke operator untuk dijadikan bahan penyelidikan di persidangan. Oleh karena itu, dibutuhkan suatu

metode dan aplikasi yang dapat mengamankan kerahasiaan informasi pada pesan SMS. (Safaat, 2013)

Beberapa penelitian yang kembangkan oleh masyarakat khususnya terkait dengan penggunaan algoritma vigenere pada SMS antara lain :

a. Widodo (2011), dengan judul Implementasi Algoritma Enkripsi Dengan Metode Modifikasi *Vigenere Cipher* dalam Aplikasi Pengiriman SMS Pada Ponsel *Blackberry*, hasil dari penelitian ini mampu menjalankan enkripsi dan diskripsi hanya pada ponsel *Blackberry* versi 5.0 kebawah.
b. Negara (2011), dengan judul Aplikasi Mobile SMS *Encryption* Menggunakan Algoritma *Shift Cipher*, hasil dari penelitian ini mampu mengenkripsi pesan sebelum dikirim dan mendeskripsi pesan yang diterima, dapat diakses melalui handphone yang sudah mendukung teknologi *Java* MIDP 2.0.
c. Dwi(2012), dengan judul Penerapan Algoritma *Vigenere Cipher* pada Aplikasi SMS *Android*, hasil penelitian ini mampu menulis pesan, mengenkripsi pesan, mengirim pesan melalui SMS, membaca pesan yang ada pada pesan masuk telepon seluler, dan mendekripsi pesan.

Berdasarkan uraian latar belakang diatas dan hasil-hasil terdahulu, peneliti bermaksud mengangkat permasalahan tersebut untuk mengembangkan aplikasi enkripsi pesan singkat pada *smartphone android* menggunakan algoritma *Vigenere* dengan memanfaatkan bilangan *euler* untuk menjaga keamanan dan kerahasiaan dari isi pesan yang dikirim.

LANDASAN TEORI

a. *Short Message Service* (SMS)

SMS (*Short Message Service*) merupakan sebuah layanan komunikasi yang ada pada telepon seluler untuk mengirim dan menerima pesan-pesan pendek. SMS pertama kali dikenalkan pada tanggal 3 Desember 1982. SMS pertama di dunia dikirimkan menggunakan jaringan GSM milik operator telepon bernama Vodafone. SMS pertama ini dikirimkan oleh ahli bernama Neil Papwort kepada Richard Jarvis menggunakan komputer. SMS dihantarkan pada channel signal GSM (*Global System for Mobile Communication*) dengan spesifikasi teknis ETSI(*European Telecomunications Standards Institute*). SMS diaktifkan oleh ETSI dan dijalankan di scope 3GPP(*3rd Generation Partnership Project*). SMS juga digunakan pada teknologi GPRS(*General Packet Radio Service*) dan CDMA(*Code division multiple access*). SMS menjamin pengiriman pesan oleh jaringan, jika terjadi kegagalan pesan akan disimpan dahulu di jaringan dan akan dikirimkan lagi ketika jaringan sudah stabil. (Dwi, 2012)

b. Algoritma

Kata algoritma berasal dari kata *algorism* yang diambil dari nama penulis buku Arab yang terkenal, yaitu Abu Ja'far Muhammad ibnu Musa al-Khuwarizmi (al-Khuwarizmi dibaca orang barat menjadi *algorism*). Al-Khuwarizmi menulis buku yang berjudul *Kitab al jabar wall-muqabala*, yang artinya "Buku pemugaran dan pengurangan" (*The book of restoration and reduction*). Dari judul buku itu kita juga memperoleh akar kata "aljabar" (*algebra*). Perubahan dari kata *algorism* menjadi *algorithm* muncul karena

kata *algorism* sering dikelirukan dengan *arithmetic*, sehingga akhiran *–sm* berubah menjadi *–thm*. Adapun pengertian algoritma adalah urutan logis langkah-langkah penyelesaian masalah yang disusun secara sistematis (Munir, 2012).

c. Kriptografi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi. (Sadikin, 2012)

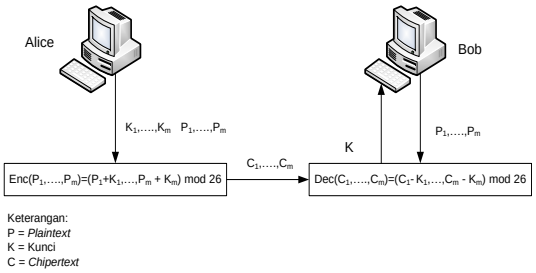
Kriptografi klasik umumnya merupakan teknik penyandian dengan kunci simetrik dan menyembunyikan pesan yang memiliki arti ke sebuah pesan yang nampaknya tidak memiliki arti dengan metode subtitusi (pergantian huruf) dan atau transposisi (pertukaran tempat). (Sadikin, 2012)

Sistem kriptografi terdiri dari 5 bagian yaitu, (Sadikin, 2012):

1) ***Plaintext* :** pesan atau data dalan bentuk aslinya yang dapat terbaca. *Plaintext* adalah masukan bagi algoritma enkripsi. Untuk selanjutnya digunakan istilah teks asli sebagai padanan kata *plaintext.*
2) ***Secret Key* :** *secret key* yang juga merupakan bagian algoritma enkripsi merupakan nilai yang bebas terhadap teks asli dan menentukan hasil keluaran algoritma enkripsi. Untuk selanjutnya digunakan istilah kunci rahasia sebagai padanan kata *secret key*.
3) ***Ciphertext* :** *chipertext* adalah keluaran algoritma enkripsi. *Chipertext* dapat dianggap sebagai pesan dalam bentuk tersembunyi. Algoritma enkripsi yang baik akan menghasilkan *chipertext* yang terlihat acak. Untuk selanjutnya digunakan istilah teks sandi sebagai padanan kata *chipertext*.
4) **Algoritma Enkripsi :** algoritma enkripsi memiliki 2 masukan teks asli dan kunci rahasia. Algoritma enkripsi melakukan transformasi terhadap teks asli sehingga menghasilkan teks sandi.
5) **Algoritma Deskripsi :** Algoritma deskripsi memiliki 2 masukan yaitu teks sandi dan kunci rahasia. Algoritma deskripsi memulihkan kembali teks sandi menjadi teks asli bila kunci rahasia yang dipakai algoritma deskripsi sama dengan kunci rahasia yang dipakai algoritma enkripsi.

d. Algoritma *Vigenere*

Sandi *vigenere* merupakan sistem sandi *poli-alfabetik* yang sederhana. Sistem sandi *poli-alfabetik* menenkripsi sekaligus sebuah teks yang terdiri dari beberapa huruf. Sandi *vigenere* menggunakan subtitusi dengan fungsi *shift* seperti pada sandi *Caesar*. Fungsi enkripsi dan dekripsi sandi *Vigenere* diberikan oleh Gambar 2.1. (Sadikin, 2012)



Gambar 2.1 Sandi Vigenere (Sadikin, 2012)

Sebagai contoh untuk menentukan nilai sandi untuk teks asli "BULANPURNAMA" dengan menggunakan sandi *vigenere* dengan himpunan kunci {7, 10, 21}. Dengan menggunakan operasi *shift* dengan himpunan kunci secara berulang dapat dihitung teks sandi sebagai berikut: (Sadikin, 2012)

c[0] = 1 + 7 mod 26 = 8
c[1] = 20 + 10 mod 26 = 4
c[2] = 11 + 21 mod 26 = 6
c[3] = 0 + 7 mod 26 = 7
c[4] = 13 + 10 mod 26 = 23
c[5] = 15 + 21 mod 26 = 10
c[6] = 20 + 7 mod 26 = 1
c[7] = 17 + 10 mod 26 = 1
c[8] = 13 + 21 mod 26 = 8
c[9] = 0 + 7 mod 26 = 7
c[10] = 12 + 10 mod 26 = 22
c[11] = 0 + 21 mod 26 = 21

Dari hasil diatas setelah di transformasi dari angka ke huruf didapatkan teks sandi : "IEGHXKBBIHWV". (Sadikin, 2012)

Keamanan sandi *Vigenere* tergantung dengan jumlah kunci yang digunakan semakin banyak jumlah kunci yang digunakan semakin luas ruang kunci. Sebagai contoh, jika jumlah alphabet adalah 26 dan sandi *vigenere* menggunakan 5 kunci maka ruang kunci adalah $26^5 \approx 10^7$. Salah satu persoalan ketika kunci yang digunakan berjumlah banyak adalah perlu mengingat kunci-kunci itu secara benar urutan dan nilainya (Sadikin, 2012).

Namun beberapa metode untuk menyerang Sandi *Vigenere* telah mengungkap kelemahan sandi ini. Analisis sandi yang diusulkan oleh Friedrich Kasiski disebut dengan pengujian Kasiski pada tahun 1863 terhadap sandi *vigenere* dapat membongkar panjang kunci dan selanjutnya membongkar nilai kunci *vigenere.* Prinsip pengujian Kasiski adalah mencari terlebih dahulu panjang kunci *vigenere* dengan mencari rangkaian karakter yang berulang (Sadikin, 2012).

e. Bilangan *Euler*

Bilangan ini adalah salah satu bilangan yang terpenting dalam matematika, sama pentingnya dengan 0, 1, *i*, dan π. Bilangan *e* yang kemudian disebut sebagai bilangan *euler* merupakan bilangan yang diperoleh dari pendekatan nilai $(1+\frac{1}{n})^n$ untuk n menuju bilangan tak terhingga, yang ditemukan pada tahun 1683 oleh Jacob Bernoulli.(Hernawati, 2009)

$$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$$

Pada tahun 1748 , Euler memberikan ide mengenai bilangan e yaitu $e = 1+\frac{1}{1!}+\frac{1}{2!}+\frac{1}{3!}+\frac{1}{4!}+\cdots$, dan bahwa $e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$. Dari formulasi tersebut, *Euler* memberikan pendekatan untuk bilangan *e* 18 digit dibelakang koma, yaitu: *e* = 2,718281828459045235. (Hernawati, 2009)

Pada tahun 1884 Boorman menghitung *e* sampai dengan 346 digit dibelakang koma dan telah dihitung sampai dengan 869.894.101 digit dibelakang koma oleh Sebastian Wedeniwski. (Hernawati, 2009)

e=2.71828182845904523536028747135266249775724709369995957496696762772407663035354759457138217852516642742746639

193200305992181741359662904357290033429526059563073813232862794349076323382988075319525101901157383418793070215408914993488416750924476146066808226480016847741185374234544243710753907774499206955170276l…… . (Hernawati, 2009)

f. Android

Android adalah sebuah sistem operasi yang berbasis *Linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang sehingga dapat menciptakan aplikasi dengan leluasa untuk digunakan oleh para pengguna *smartphone android*. (Safaat, 2012)

g. *Android Sofware Development Kit* (*Android* SDK)

Android software development kit adalah *tool application programming interface* yang diperlukan untuk memulai pengembangan aplikasi pada platform *android* mengunakan bahasa pemograman *java*. (Safaat, 2012)

h. UML

UML (*Unified Modeling Language*) adalah bahasa yang menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. (Sugiarti, 2013)

Menurut Sugiarti (2013), UML menyediakan beberapa diagram untuk memodelkan aplikasi berorientasi obyek, diantaranya adalah *Use Case Diagram, Activity Diagram, Class Diagram, Sequence Diagram, Collaboration Diagram, Component Diagram, Statechart Diagram, dan Deployment Diagram.*

ANALISA DAN PERANCANGAN

3.1. Analisa Sistem

Kegiatan analisa sistem adalah kegiatan untuk melihat sistem yang sudah berjalan, melihat bagian mana yang bagus dan tidak bagus, dan kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru. (Rosa dan Shalahuddin, 2011)

3.2. Identifikasi Awal

Identifikasi merupakan urutan kegiatan yang tepat dari tahapan-tahapan yang menerangkan mengenai proses yang dikerjakan, bagaimana proses dapat dikerjakan dan dokumen yang dilibatkan.

3.3. Analisa Kebutuhan Perangkat Lunak

Analisa kebutuhan perangkat lunak dalam mengembangkan aplikasi enkripsi pada pesan singkat terdiri atas analisa tentang kebutuhan fungsional, dan kebutuhan non fungsional.

3.3.1. Kebutuhan Fungsional

Analisa kebutuhan fungsional membahas tentang kebutuhan yang harus dikerjakan oleh aplikasi. Berikut ini merupakan kebutuhan fungsional pada aplikasi enkripsi pada pesan singkat:

a. Enkripsi pesan sebelum dikirim.
b. Mengirim pesan.
c. Menerima pesan.
d. Dekripsi pesan masuk.

3.3.2. Kebutuhan Non Fungsional

Analisa kebutuhan non fungsional membahas tentang kebutuhan perangkat yang digunakan dalam pengembangan aplikasi enkripsi pesan singkat yaitu :

a. Analisa Kebutuhan Perangkat Lunak (*Software*)

Kebutuhan perangkat lunak yang diperlukan dalam mengembangkan aplikasi enkripsi pada pesan singkat adalah sebagai berikut:

1) *Eclipse* dengan ADT *plugin*.
2) *Android* SDK.

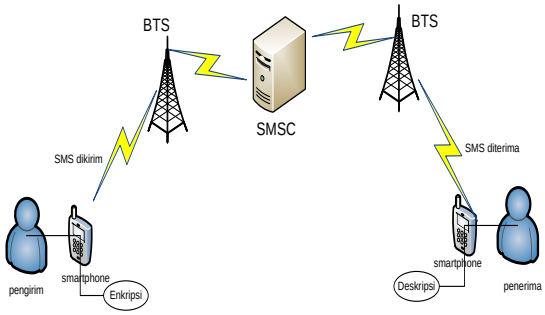
b. Analisa Kebutuhan Perangkat Keras (*Hardware*)

Perangkat *smartphone* yang digunakan untuk mengoperasikan aplikasi enkripsi pada pesan singkat sebagai berikut :

2) *Processor* 832 MHz.
3) Memori RAM 290 MB.
4) Memori Internal 160 MB.

3.4. Perancangan Arsitektur

Rancangan arsitektur aplikasi SMS *Vigenere* dapat di lihat pada gambar 3.1:



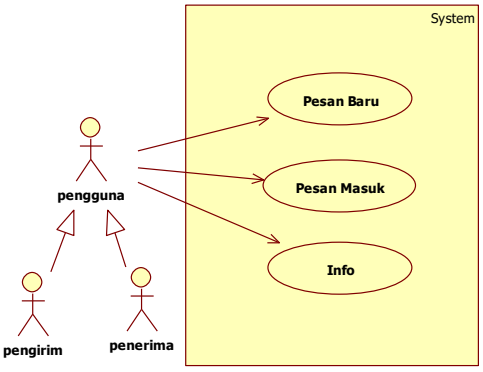
Gambar 3. 1 Arsitektur Pengiriman SMS Vigenere

3.5. Perancangan Model Proses

Rancangan aplikasi enkripsi pesan singkat menggunakan rancangan UML yang terdiri dari : *Use Case Diagram*, *Class Diagram*, *Activity Diagram* dan *Sequence Diagram.*

a. Use Case Diagram

Use Case Diagram memperlihatkan bagaimana peran setiap aktor dalam interaksi dengan sistem. *Use case diagram* untuk aplikasi enkripsi SMS yang akan dikembangkan dapat dilihat pada gambar 3.2 :



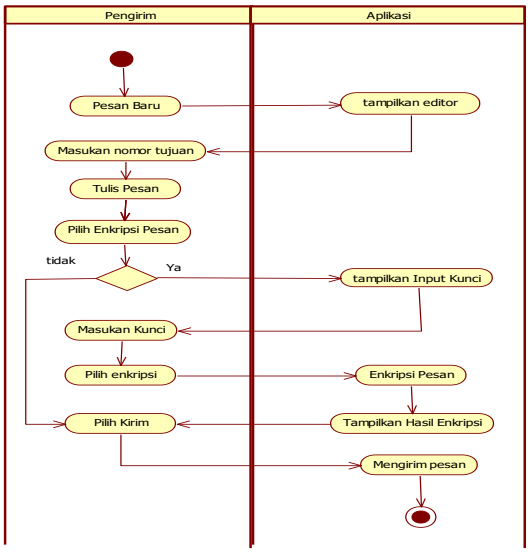
Gambar 3. 2 Use case diagram SMS Vigenere

b. Activity Diagram

Aplikasi SMS *Vigenere* memiliki rancangan *activity diagram*. Adapun *activity diagram* aplikasi SMS *Vigenere* adalah sebagai berikut :

1) *Activity Diagram* Pesan Baru

Activity diagram pesan baru dapat dilihat pada gambar 3.3:

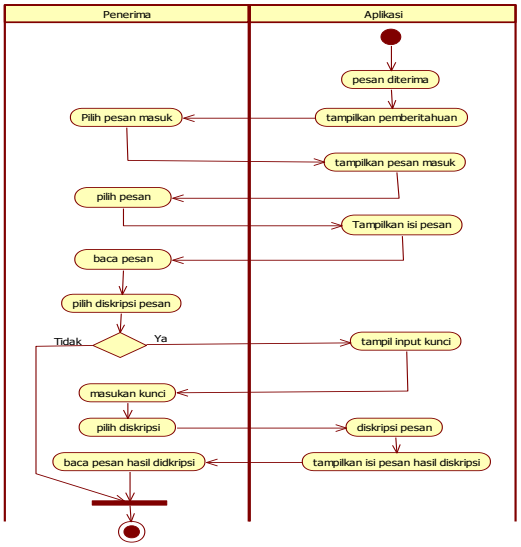


Gambar 3. 3 Activity Diagram Pesan Baru

Gambar 3.3 merupakan alur proses saat pengirim melakukan tulis pesan, enkripsi dan mengirim pesan.

2) *Activity Diagram* Pesan Masuk

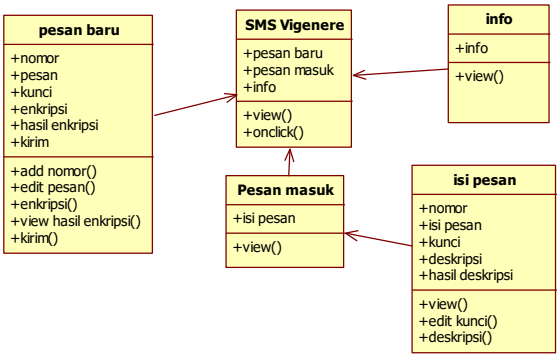
Activity diagram pesan masuk dapat dilihat pada gambar 3.4:



Gambar 3. 4 Activity Diagram Pesan Masuk

c. *Class Diagram*

Class diagram SMS *Vigenere* dapat dilihat pada gambar 3.5



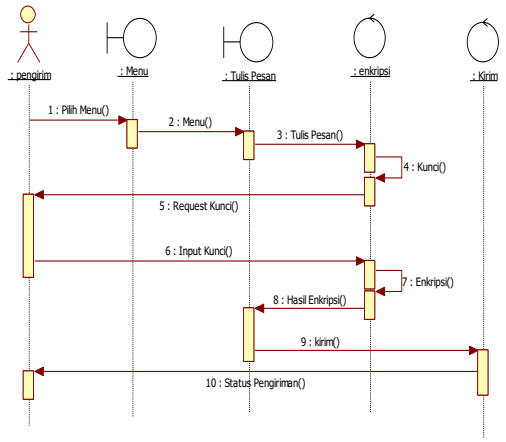
Gambar 3. 5 Class diagram SMS Vigenere

d. *Sequence Diagram*

Aplikasi SMS *Vigenere* memiliki rancangan *sequence diagram*. Adapun beberapa *sequence diagram* aplikasi SMS *Vigenere* adalah sebagai berikut :

1) *Squence Diagram* Pesan Baru

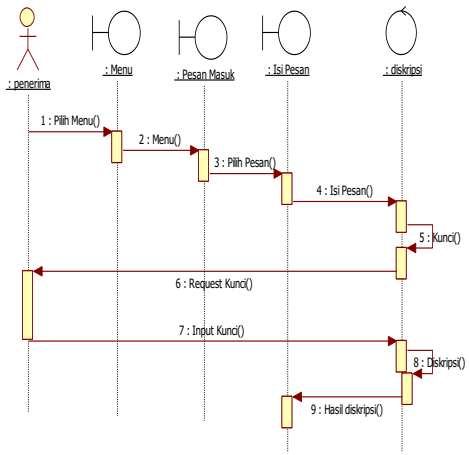
Squence diagram pesan baru dapat dilihat pada gambar 3.6:



Gambar 3. 6 Squence Diagram Pesan Baru

2) *Squence Diagram* Pesan Masuk

Squence diagram pesan masuk dapat dilihat pada gambar 3.7 :

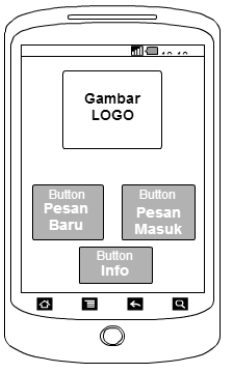


Gambar 3. 7 Squence Diagram Pesan Masuk

3.6. Perancangan Antar Muka

Perancangan antar muka merupakan bentuk rancangan yang akan menghubungkan aplikasi dengan pengguna yang berkaitan dengan aplikasi SMS *Vigenere.* Antar muka halaman menu merupakan tampilan awal dari

aplikasi SMS *Vigenere*. Adapun antar muka halaman menu dapat dilihat pada gambar 3.8.



Gambar 3. 8 Antar muka halaman menu

IMPLEMENTASI DAN PENGUJIAN

Implimentasi

Implementasi merupakan tahap pengembangan aplikasi enkripsi pesan singkat. Pada tahap ini, perancangan yang sudah dibuat secara konsep mulai diterapkan ke dalam rancangan yang sebenarnya.

Pembahasan

Untuk melakukan enkripsi dan dekripsi menggunakan *vigenere* dengan memanfaatkan digit bilangan *euler* dibutuhkan plainteks, kunci dan bilangan *euler*. Pada enkripsi dan dekripsi dengan memanfaatkan bilangan *euler* dapat dilihat pada fungsi dibawah ini:

$$Ci = ((Pi+BEi)+Ki) \bmod 95$$

Secara matematis cipherteks dinotasikan Ci, plainteks dinotasikan Pi, digit bilangan *euler* dinotasikan BEi dan kunci dinotasikan Ki. Dari fungsi tersebut dapat diuraikan algoritma menggunakan *vigenere* yang memanfaatkan digit bilangan *euler* yaitu:

a. Plainteks(Pi) dikonversi ke dalam angka yang mewakili huruf, angka dan tanda baca lain yang terdapat dalam 256 kode ASCII dalam penelitian ini menggunakan *range* antara 32-126 yaitu terdapat 95 karakter.
b. Digit bilangan euler (BEi) dikonversi ke dalam angka dengan kode ASCII seperti plainteks. Pada penelitian ini menggunakan 160 digit pertama dari bilangan euler.
c. Kunci (Ki) dikonversi ke dalam angka dengan kode ASCII.
d. Menentukan panjang digit bilangan euler sesuai panjang plainteks.
e. Penjumlahan plainteks dengan digit bilangan euler. Dari hasil penjumlahan tersebut dijumlahkan dengan kunci. Cek jika kunci kurang dari plainteks kunci akan melakukan looping sampai sama panjang dengan plainteks. Hasil penjumlahan tersebut kemudian dimodulo 95. Kemudian hasilnya di konversi dalam bentuk karakter. Hasil akhir tersebut adalah cipherteks(Ci).

Dibawah ini merupakan contoh dalam enkripsi menggunakan algoritma *vigenere* yang memanfaatkan digit bilangan *euler*.

Plainteks = Universitas

Bilangan euler = 27182818284 (panjang digit yang dipakai disesuaikan dengan panjang plainteks).

Kunci = respati

Tabel 4. 1 Penjumlahan plainteks dengan bilangan euler

Plainteks	U	n	i	v	e	r	s	i	t	a	s
kode ASCII	85	110	105	118	101	114	115	105	116	97	115
Bil.euler	2	7	1	8	2	8	1	8	2	8	4
kode ASCII	50	55	49	56	50	56	49	56	50	56	52
ASCII	135	165	154	174	151	170	164	161	166	153	167
hasil mod 95	40	70	59	79	56	75	69	66	71	58	72

Ubah kunci ke dalam bentuk angka dengan kode ASCII. Kemudian masukan hasil penjumlahan plainteks dengan euler dan kunci ke rumus *vigenere* yang sebelumnya masing-masing angka dikurangi 31 terlebih dahulu karena panjang *range* yang digunakan 95. Berikut ini adalah tabel penjumlahan

Tabel 4. 1 Penjumlahan Vigenere

hasil Pi+BEi	40	70	59	79	56	75	69				
(-)31	9	39	28	48	25	44	38				
kunci	r	e	s	p	a	t	i				
kode ASCII	114	101	115	112	50	116	105				
(-)31	83	70	84	81	151	85	74				
hasil (+)	92	109	112	129	176	129	112				
mod 95	92	14	17	34	81	34	17	23	15	16	27
hasil (+)31	123	45	48	65	112	65	48	54	46	47	58

Dari hasil diatas, masing-masing angka dirubah ke dalam bentuk karakter sesuai kode ASCII, berikut adalah hasil cipherteks : {-0AzA06./:

4.1. Pengujian

Pengujian ini dilakukan dengan mengirimkan pesan pada nomor sendiri, sehingga dapat dibaca kembali isi pesan yang diterima.

a. Pengujian Enkripsi Pesan

Pada pengujian enkripsi pesan langkah yang dilakukan yaitu tulis pesan kemudian pilih enkripsi pesan dan masukan kunci kemudian tekan enkripsi. Berikut ini adalah hasil pesan yang belum terenkripsi sampai dienkripsi.

Gambar 4. 1 Hasil enkripsi pesan

Berikut ini merupakan cuplikan source code untuk proses enkripsi pesan menggunakan algoritma *vigenere* dengan memanfaatkan bilangan *euler*.

```
        protected void enkrip()
{
        Skunci=kunci.getText().
toString();
        Spesan=pesan.getText().
toString();
        String cipherText = "";

        String       bil_euler=
"271828182845904523536028747135266249775724709369995957496696
```

```
      676277240766303535475945713821785251664274274663919320030059
      921817413596629043572900334295260595630738132328627943490 7";
      if (Skunci.length()==0){
      Toast.makeText(getBaseContext(),"Kunci Tidak Boleh
      Kosong!!", Toast.LENGTH_SHORT).show();}
      else if (Skunci.length()>0 &&
      Spesan.length()>0) {
      int panjang_row =95;
      int panjang_kolom = 95;
      int tabel_vignere[][]= new int[95][95];
      for (int rows = 0; rows < panjang_row; rows++){

      for(int kolom = 0; kolom < panjang_kolom; kolom++){
      tabel_vignere[rows][kolom]= (rows+kolom)%95; }}
      int keyIndex = 0;
      int Indexeuler = 0;
      for (int ptextIndex = 0;ptextIndex <Spesan.length();
      ptextIndex++){
      pChar = Spesan.charAt(ptextIndex);
      int asciiVal = (int) pChar;
      if (asciiVal < 32 || asciiVal > 126){
      cipherText += pChar;continue;}
      int PlainTextValue = ((int) pChar);
      char val=bil_euler.charAt(Indexeuler);
      int tes=(int)val;
      int Text=(PlainTextValue+tes)%95;
      Indexeuler++;
      if (Indexeuler == bil_euler.length())
      Indexeuler = 0;
      int plain=Text-31;
      char kChar = Skunci.charAt(keyIndex);
      int KeyValue = ((int) kChar ) - 31;
      int tableEntry = tabel_vignere[plain][KeyValue];
      char cChar = (char) (tableEntry + 31);
      cipherText += cChar;
      keyIndex++;
      if (keyIndex == Skunci.length())
      keyIndex = 0;}
      hasil_enkrip.setText(cipherText)}
      else{Toast.makeText(getBaseContext(),"Pesan Tidak Boleh Kosong!!", Toast.LENGTH_SHORT).show();}}
```

b. Dekripsi pesan

Pada pengujian ini yang dibutuhkan yaitu kunci yang sama seperti pada proses enkripsi. Berikut ini merupakan hasil dari dekripsi pesan yang sudah terenkripsi sebelumnya. Pesan masuk yang terenkripsi dapat di lihat pada gambar 4.2 dan gambar 4.3 merupakan hasil pesan yang sudah di deskripsi, yaitu sebagai berikut:

Gambar 4. 2 Isi pesan terenkripsi

Gambar 4. 3 hasil pesan yang di dekripsi

Selama dilakukan pengujian dengan mengirim pesan terenkripsi secara berulang-ulang dengan isi pesan dan kunci yang berbeda-beda, tidak ditemukan masalah. Semua fungsi dapat berjalan dengan baik.

PENUTUP

Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya, serta proses pengujian terhadap aplikasi yang telah dibuat, maka dapat di tarik beberapa kesimpulan, yaitu sebagai berikut:

a. Layanan SMS dapat mengirim dan menerima pesan yang bersifat personal atau rahasia dari orang lain namun yang dikirimkan melalui SMS tidak menjamin keamanan dan kerahasiaannya. Karena pesan yang dikirim menggunakan aplikasi SMS bawaan ponsel masih berupa teks terbuka yang belum terproteksi. Sehingga pesan yang bersifat personal atau rahasia tidak dijamin kerahasiaannya sampai ke penerima tanpa dicuri informasinya oleh orang lain. Dengan menenkripsi pesan SMS sebelum pengiriman dengan menggunakan algoritma *Vigenere* yang memanfaatkan digit bilangan *euler* menjadi lebih acak sehingga akan menyulitkan kriptanalisis untuk menyerang.
b. Untuk dapat membaca makna dari pesan tersebut, penerima pesan perlu melakukan dekripsi isi pesan tersebut menggunakan kunci yang sama yang digunakan oleh pengirim. Apabila ada orang lain yang mencuri isi pesan tersebut, orang tersebut tidak akan mampu membaca makna pesan tersebut karena dalam kondisi terenkripsi. Dengan adanya sistem keamanan ini isi pesan yang bersifat personal atau rahasia dapat tersampaikan secara aman.

Saran

Beberapa saran-saran yang perlu diperhatikan untuk pengembangan aplikasi enkripsi SMS menggunakan algoritma *vigenere* dengan memanfaatkan digit bilangan *euler*, yaitu sebagai berikut:

a. Aplikasi ini dapat dikembangkan dengan menambahkan pemberitahuan jika ada pesan masuk.
b. Aplikasi ini dapat dikembangkan dengan menambahkan tanda untuk pesan masuk yang belum dibaca.
c. Untuk kedepannya dalam memanfaatkan bilangan *euler* gunakan perkalian agar cipherteks yang dihasilkan lebih acak sehingga dapat menyulitkan kriptanalis dalam menemukan panjang kunci.

DAFTAR PUSTAKA

Dwi, Andi Kurniawan. 2012. *Penerapan Algoritma Vigenere Cipher pada Aplikasi SMS Android*. http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2011-2012/akalah2012/Maka_lah-Kripto-2012-031.pdf, diakses pada tanggal 3 Juni 2014 jam 21.30.

Hernawati, Kuswari. 2009. *Implementasi Cipher Viginere pada kode ASCII dengan Memanfaatkan Digit Desimal Bilangan Euler*. http://staff.uny.ac.id/sites/default/files/penelitian/Kuswari%20Hernawati,%20S.Si.,M.Kom./Bilangan%20Euler%20pada%20Kriptografi.pdf, diakses pada tanggal 3 Juni 2014 jam 20.00.

Negara, Asep Pristia. 2011. *Aplikasi Mobile SMS Encryption Menggunakan Algoritma Shift Cipher*. http://repository.amikom.ac.id/files/Publikasi 07.11.1731.pdf, diakses pada tanggal 3 Juni 2014 jam 20.30.

Rizkiansyah, Irvan. 2013. *Pengembangan Aplikasi Pembelajaran Interaktif Teknik*

Bermain Piano Berbasis Multimediadi Lembaga Kursus Musik Ethnictro Yogyakarta. http://eprints.uny.ac.id/10031/1/JURNAL.pdf, diakses pada tanggal 22 Agustus 2014 jam 15.00.

Widodo, Dyan Hari. 2011. *Implementasi Algoritma Enkripsi Dengan Metode Modifikasi Vigenere Cipher Dalam Aplikasi Pengiriman SMS pada Ponsel Blackberry.* http://repository.amikom.ac.id/files/Publikasi07.11 .1471.pdf, diakses pada tanggal 3 Juni 2014 jam 20.00.

Munir, Rinaldi. 2012. *Matematika Diskrit.* Informatika: Bandung.

Sadikin, Rifki. 2012. *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta : Andi Offset.

Safaat, Nazruddin H. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Bandung:Informatika.

Safaat, Nazruddin H. 2013. *Aplikasi Berbasis Android.* Bandung : Informatika.

Shalahudin, M dan S, Rosa A. 2011. Rekayasa Perangkat Lunak(Terstruktur dan Berorientasi Objek). Modula : Bandung.

Sugiarti, Yuni. 2013. *Analisis & Perancangan UML (Unified Modeling Language) Generated VB.6*. Yogyakarta : Graha Ilmu.